The NCEM Image Processing Custom Function Plug-Ins for Digital Micrograph

Roar Kilaas, Sidnei Paciornik and John Bonevich

User Manual

This package has been created by Dr. Roar Kilaas, Dr. Sidnei Paciornik and Dr. John Bonevich at the National Center for Electron Microscopy at the Lawrence Berkeley National Laboratory. We do try to ensure that the software is free of errors, but we cannot guarantee the software in any way. We use these routines in our own research so we do make efforts to ensure that the routines work as well as possible. We hope they can be of use to others as well.

This work is supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Materials Sciences Division of the U.S.Department of Energy under Contract No. DE-AC03-76SF00098.

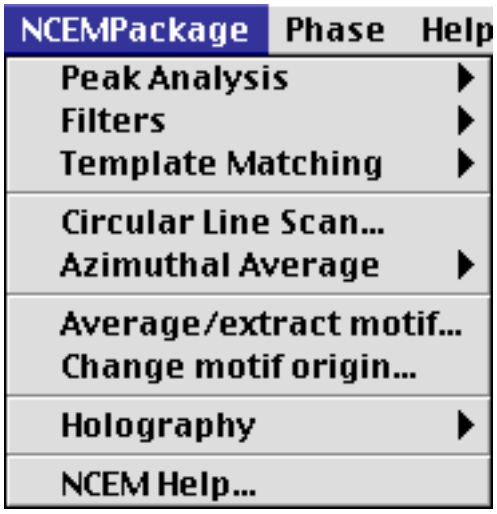
For communication on the routines, our addresses are as follows:

Roar Kilaas: (software development and implementation)
email: R_Kilaas@lbl.gov
Phone: (510) 486-4618

Sidnei Paciornik: (help system and testing)
email: Sidnei@dcmm.puc-rio.br

John Bonevich (holography software development and implementation)
email: jebonevich@nist.gov
Phone: (301) 975-6139

The NCEM Package Menu Options



The package installs by placing the routines in the “Packages” folder that resides in the same location as the Digital Micrograph application. Once the program is started a new menu shows up in the menubar. The items in the menu are shown above and some of the menuitems point to sub-menus.

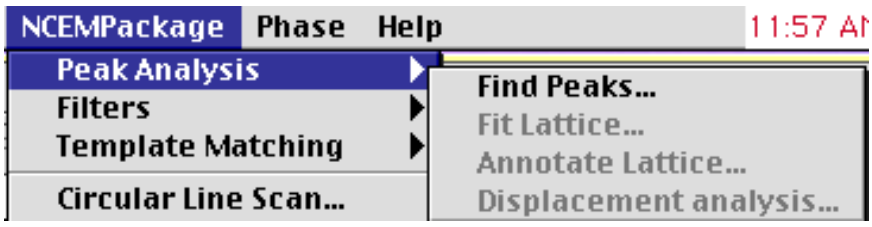
Operating the routines

Operating the routines

The following sections outline the use of the routines. Each menu-item is given its own section and the options and input/output for each routine is described in detail.

Peak Analysis Sub Menu

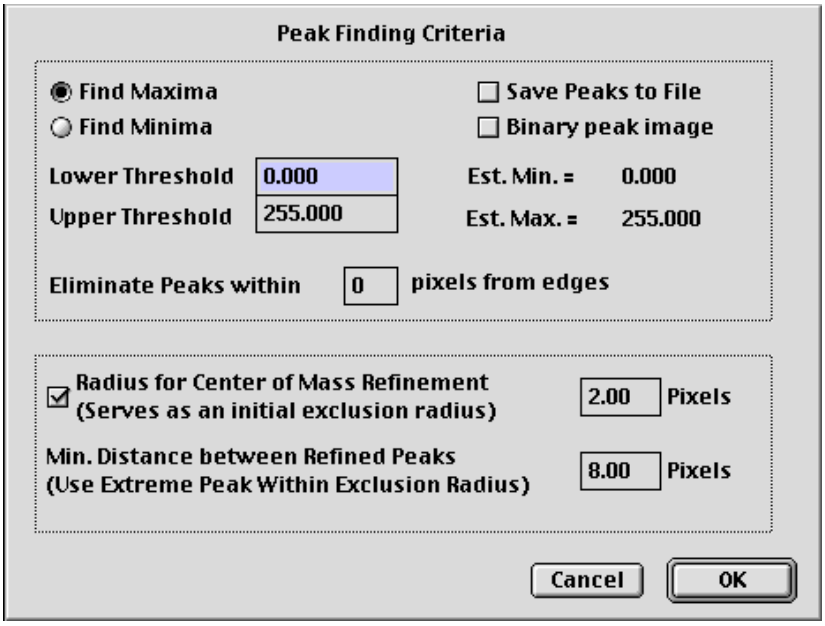
Peak Analysis



Find Peaks

Find Peaks....

This routine is used to locate local maxima (or minima) in the active image or a selection therein. It is one of the most useful routines for the analysis of HREM images. It's results are used by other plug-ins like Fit Lattice, List Displacements and Unit Averaging.



It uses the following parameters:

- Lower Threshold
- Upper Threshold
- Distance from the Edge
- Radius for Center of Mass Refinement
- Minimum Distance between Peaks

The procedure works as follows:

1 It first locates all pixels which are local maxima (or minima) in 3x3-pixels neighborhoods, that have intensities between the Lower and Upper Thresholds and which are no closer to the edges than the minimum Distance from the Edge. This will find both isolated peaks and

"plateaus" of adjacent peaks. The number of unrefined peaks is displayed in the Results window.

2 It then reduces plateaus to single peaks. The program first creates a list of all plateaus and the pixels contained within each plateau. It then takes all the pixels belonging to the plateau and finds the pixel that lies closest to the center of mass of the plateau.

3 If the Radius for Center of Mass Refinement (RCM) is different from zero, each peak found in the previous steps is refined to sub-pixel accuracy. This is done by calculating the center of mass of intensities in a circular region within RCM from each peak. The larger RCM the more accurate is the refinement. However RCM should be no bigger than half the distance between nearest neighbor peaks or else one peak will influence the other and produce wrong results. If the circular region touches the borders of the image, the peak is left unrefined. RCM is also used as an initial exclusion radius (see below) to refine a noisy peak (composed of various sub-peaks) to a single peak. It takes the brightest pixel within RCM as the central peak for center of mass refinement and ignores all other sub-peaks. The best value for RCM is normally the radius of an image dot.

4 If the Minimum Distance between Peaks (RMIN) is different from zero, the program will ignore any peaks within RMIN of a previously refined peak. This feature is best used in combination with the refinement provided by RCM. The best value for RMIN is normally the distance between "true" peaks.

Once the peaks are found and refined the program writes the number of refined peaks in the Results window and presents a copy of the image with the peaks marked as crosses. In this version the crosses are actually written on the image and not just annotations. When finding maxima (minima) the crosses will be written with the minimum (maximum) value of the image to provide good contrast. However, the central spot of each cross preserves the original pixel value for easy comparison.

The program automatically saves the x and y coordinates of each peak as pixels in a standard Digital Micrograph image. This image has two rows and a number or columns equal to the number of peaks. The first row contains the x coordinate and the second the y coordinate of the peaks. This image has the fixed name "PeakList" and is kept hidden. It is used by other plug-ins like Fit Lattice which looks for an image with this name. The Peak

Finding routine will overwrite this image every time it is run. To preserve it, the user must save it with a different name.

The user also has the option of saving a data file that contains the peak list. The data file can have one of three formats:

1 Standard Text Format which is a simple text file with four columns separated by tabs and a number of rows equal to the number of peaks found. The first and second columns contain the x and y coordinates respectively. The third contains the intensities. The fourth contains either the total summed intensities in the center of mass region, if the peak was refined, or just the intensities (as in the third column) if the peak was not refined. The default name for this file is the name of the image with the suffix ".pk". The data is sorted in ascending y order.

2 MacTempas Format which is a text file that can be read by MacTempas as a standard atomic position and microscope information list. As at this stage the Peak Finding routine has no information about the kinds of atoms and microscopes used, it simply writes default data to make the file readable by MacTempas. It is the responsibility of the user to change this information later. For the data file to make any sense it is necessary to have a scale set for the image (if no scale is set the program will warn the user only the first time it is run for a given image). The default name for this file is the name of the image with the suffix ".at".

3 Semper Format which is a binary file that can be read by Semper through its import command. The final file name is the name chosen by the user with the number of peaks added to the end. Semper must then reclass the image to be of the PList type before it can access the data in a meaningful way. If the data is to be read by Semper running on a PC, then one must take care to swap bytes when importing the file and restrict the file name to the PC standard.

If there is a selection in the image the program locates peaks only within it and any edge related procedures will then refer to the edge of the selection. In this version there is not attempt to map the coordinates inside the selection, which have as origin the top left corner of the selection, to the coordinates of the original image.

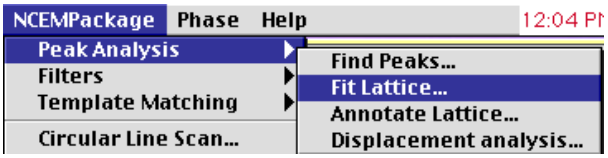
The program correctly locates peaks in one-dimensional images like, e.g., line profiles. However, the display of the results can be confusing because of the way the program marks the peak positions. Unfortunately Digital

Micrograph does not currently allow any form of annotation on one-dimensional images.

During processing the program creates a temporary Scratch image, which is later deleted, and a Cross Image, as described above. If the program runs out of memory it displays a message saying “Unable to Create Scratch (or Cross) Image”.

Fit Lattice

Fit Lattice



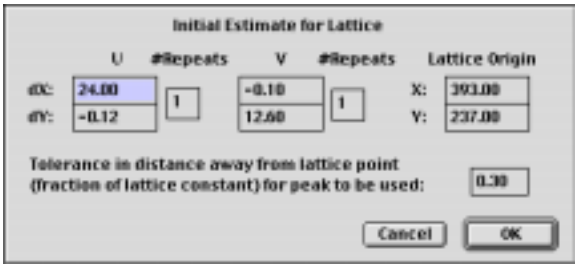
This routine calculates a 2-dimensional lattice fit to a set of (x,y) positions. It generates a best fit in the least-squares sense.

It takes the following input:

i) the PeakList image, created by the Find Peaks routine, which contains the (x,y) coordinates of the peaks in an image. The program looks for an image with this name and will present an error if it is not present. Please read the help for the Find Peaks routine for more information.

ii) A first estimate for the lattice vectors and origin. This estimate is created with the **Lattice Tool** . See description below.

iii) A tolerance, in fraction of the lattice parameter, for including peaks in the fitting procedure. The smallest this value, the strictest is the fit and less peaks are considered. However, this can lead to a erroneous



fit which includes too few peaks.

It creates the following output:

i) The refined u and v lattice vectors and origin w.

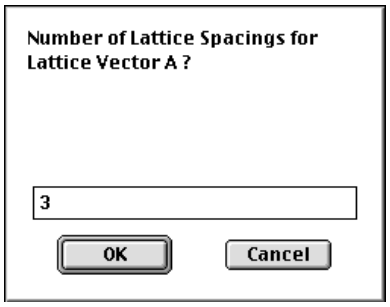
ii) The number of peak positions actually used in the fitting procedure.

iii) A fitting error estimate based on the Root Mean Square (rms) difference between the fitted lattice points and the peak positions used in the fitting.

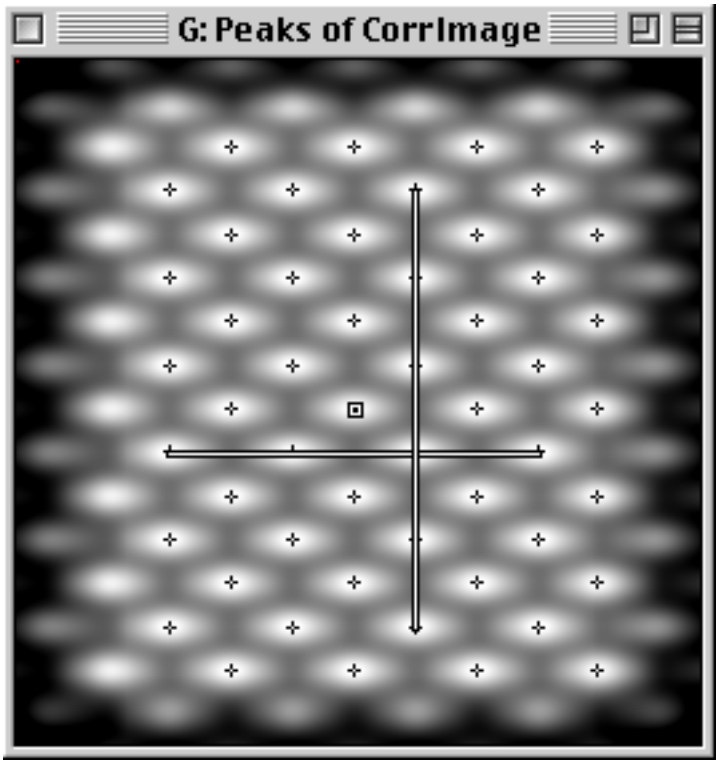
The Lattice Tool

This tool is used to provide, through a graphical interface, a first estimate of the lattice vectors and origin. After running Find Peaks and obtaining a Cross Image (see help for Find Peaks) the user selects the Lattice Tool from the ToolBox. The sequence is then as follows:

- Using the mouse, and the crosses of the Cross Image as reference, draw a line spanning a given number of lattice parameters along a given direction. The program then asks for the number of repeats and obtains an estimate for the lattice parameter to be used later. Due to some quark of the code, the first drag produces nothing and a second drag is needed to define the first lattice parameter.



- Repeat the procedure above for the other lattice direction.
- Using the mouse, click on a position to be used as the lattice origin. The program marks this position with a square. The square can be moved to change the origin.

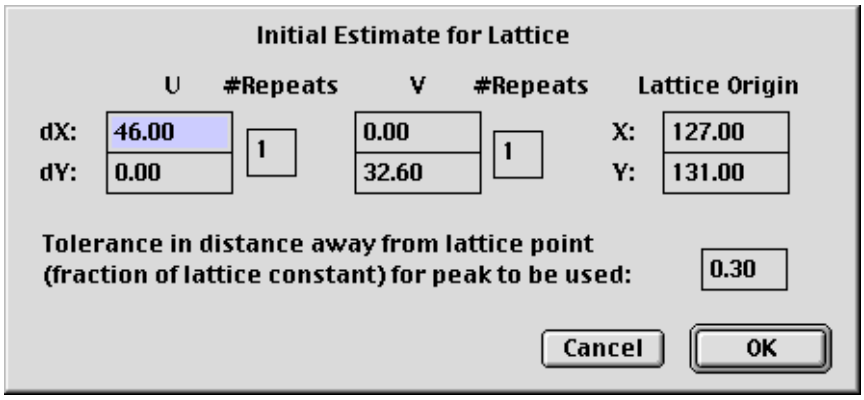


Important Note : There exists a potential problem with the use of the annotations (in association with its window) created by the Lattice Tool. As originally implemented, the annotations were deleted (hidden) when the window was deactivated and/or another tool selected. This was done so that the annotations would not accidently be moved or changed outside the use of the Lattice Tool. However, due to a persistent problem associated with closing the window with the Lattice Tool selected (DM sends a select/deselect tool message to the tool when the window no longer exists), the program would crash. Instead of risking that the user looses valuable data, this has been changed so that now the annotations exist after the Lattice Tool has completed its mission and they take on a life of their own. This information is to warn the user that some “strange”, but not harmful, behavior may be observed because of the persistence of lattice annotations outside the scope of the Lattice Tool.

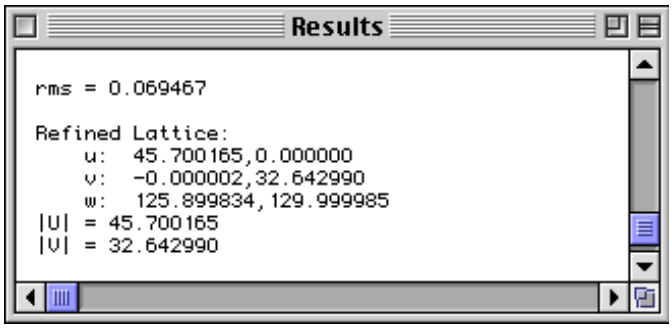
Running Fit Lattice

When the program is called, it puts up a dialog box containing the first estimates for the lattice vectors and origin created with the Lattice Tool, and

the tolerance. These values can be changed by the user but in general there is no need to do so.



The program then calculates the best fit and outputs, in the Results Window, the number of peaks used, the rms error and the coordinates of the lattice vectors and the origin. It also creates the Lattice image, which contains the lattice vectors and origin coordinates, to be used by other routines (e.g. List Displacements).



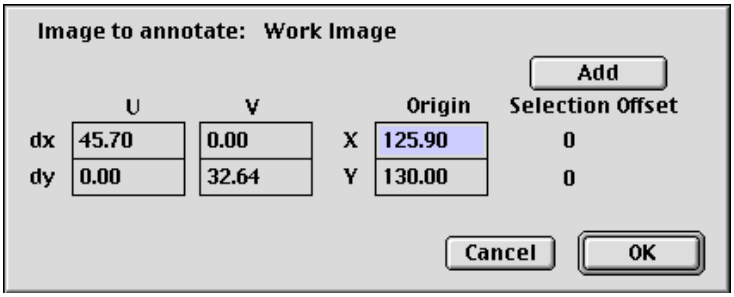
The accuracy of the fit can be improved by running Fit Lattice recursively. The program automatically takes the previous results as input for the next run. In principle, this procedure should be repeated until the rms value reaches a stable value.

Annotate Lattice

Annotate Lattice

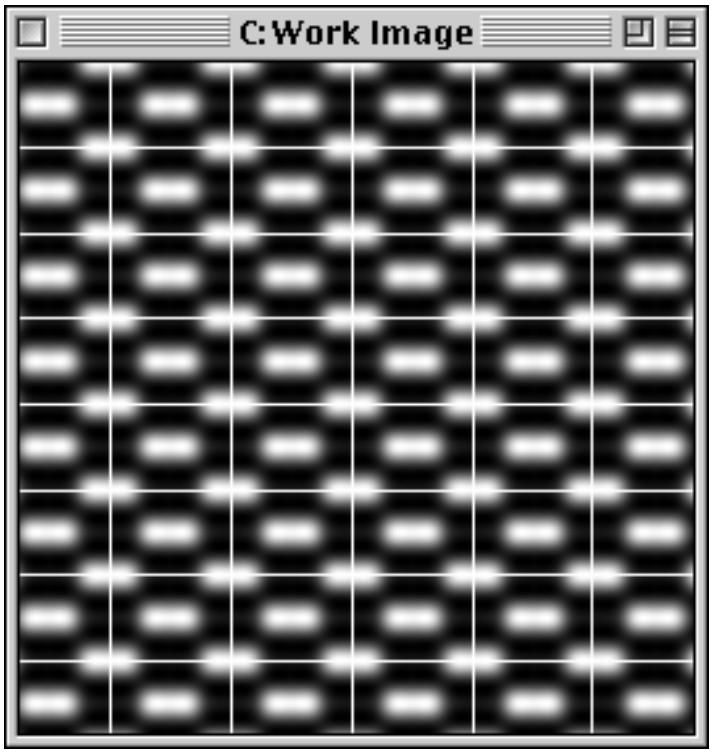
This routine draws a set of line annotations on top of the front image, to display graphically the result of the Fit Lattice routine.

If the lattice was fitted on a selection of the original image, and the user wants to annotate the whole image, the program offers the option of adding the top-left coordinates of the selection so that the annotate lattice routine will correctly use the top-left coordinate of the whole image as the origin.



Unfortunately, due to a bug in the current version of Digital Micrograph, this procedure sometimes presents erroneous behavior as follows:

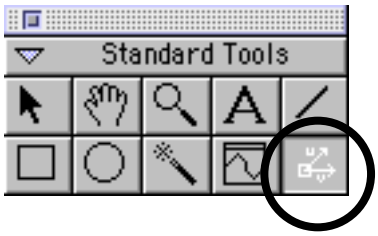
In certain instances, the display window is not updated correctly and the lattice annotation looks wrong. However, forcing a screen update (e.g. by dragging another window on top of the annotated one and then dragging it away) should take care of this problem.



Important Note : When the number of annotations is large (and we do not know exactly how much is “large”), the program might (will?) crash so you probably should not try to annotate a lattice on an image much larger than 200 by 200 pixels. In any case, before doing so, you should save your work.

Lattice Tool

Lattice Tool



Lattice Tool is a tool installed in the “Tools Palette” under the custom tools. For a description on how to use this tool, see help on Fit Lattice.

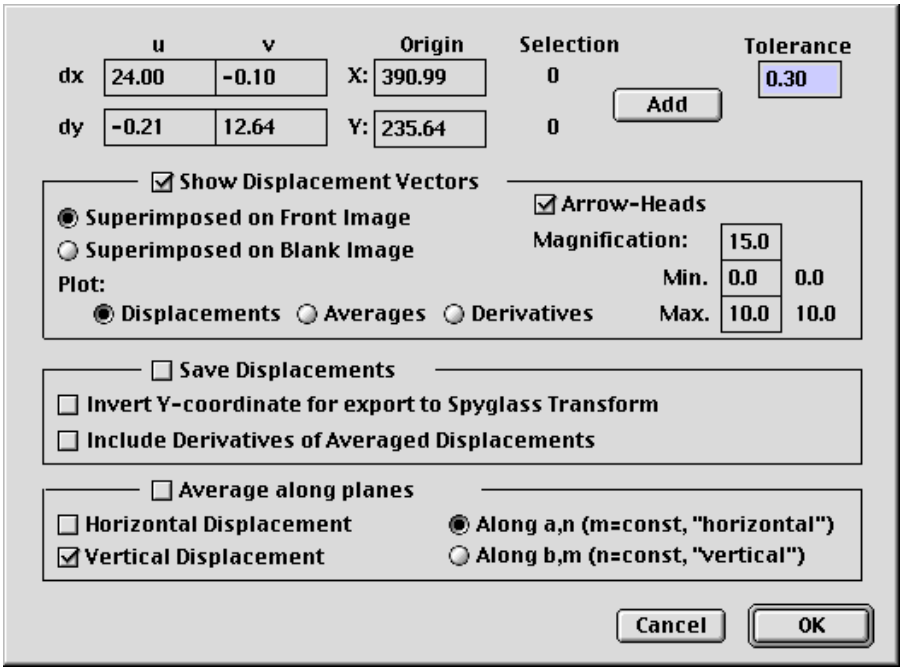
Displacement Analysis

Displacement Analysis

This routine calculates vector displacements between a fitted lattice parameters list, created by the Fit Lattice routine and the peak list created by the Find Peaks routine. The two lists can come from the same image or from different images.

In the current version, the routine looks for these lists in images with the specific names (“Lattice” and “PeakList”) with which they are created by the other routines, and will not accept other names. If the user desires to use lists which have been saved as images with different names, they must be renamed first to match these names. It must be noted, however, that if another Lattice or PeakList image already exists at this point, Digital Micrograph will accept setting other images with these same names but the List Displacements program will choose one of them, and it is not possible to predict which one. It is highly advisable that this kind of name changing be done very carefully.

The routine takes the following input:



- - The lattice vectors u and v and the origin w. The default values for these values are obtained automatically from the Lattice image.

- - The ADD/SUBTRACT selection coordinates option. If the Lattice fit was done on a selection of a given image, and the user wants to calculate displacements for the whole image, the program offers the option of adding/subtracting the top-left coordinates of the selection so that the list displacements routine will correctly use the top-left coordinate of the whole image as the origin. The button will toggle between ADD and SUBTRACT. If this option was active in the Annotate Lattice routine the program will show the coordinates already added and the button will show SUBTRACT.

- - A tolerance, in fraction of the lattice parameter, for including peak positions in the displacement calculation. Peaks which are farther away from a fitted lattice position than the tolerance are excluded.

It creates the following output:

- A vector map of the displacements which is drawn either on top of the front image or on a new blank image.

The user can select the range of displacement values to be shown and a magnification factor between the actual displacement and the annotation length. Outside the min and max range boxes the program presents the global min and max for the displacement vector magnitudes. However, the first time the program is called on a given image, it does not have the correct values for these parameters. The user should then run it once with the default values and then, when running it again, the correct min and max values will be shown. Because Digital Micrograph does not allow control over the size of the arrowheads, which for small displacements can be too large, the user can select to show the annotations as plain lines. Instead of displaying the displacements, the user can choose to show the average displacement or the derivative of the average displacement.

These options are only active if the Average box is clicked at the bottom of the dialog box. See description below.

- A text file with the results of the displacement calculation. This file is composed of a number of columns that depends on the options clicked (see below) and as many rows as the number of positions included in the calculation.
- Columns 1 and 2 contain the x and y coordinates of the fitted lattice positions.
- Columns 3 and 4 contain the x and y coordinates of the original peaks.
- Columns 5, 6 and 7 contain, respectively, the x-displacement, the y-displacement and the resulting r displacement.
- Columns 8 and 9 contain the index numbers (n,m) of each position in relation to the **u** and **v** lattice vectors and the lattice origin w. Thus, Peak(n,m) = w + n**u** + m**v**
- If the average option is clicked at the bottom of the dialog box (see below) and also in the save file section, columns 10 and 11 will contain the averages for x and y respectively.
- If the average option is clicked at the bottom of the dialog box (see below) and the derivative option is clicked in the save file section, columns 12 and 13 will contain the derivatives for x and y respectively.

The columns are separated by tabs and the file can be read by any plotting application. In particular, the user might want to use Spyglass Transform to create the vector plot because it provides more options in terms of the graphical output. In this case the user must click the option “Invert Y-coordinate for export to Spyglass Transform”.

At the bottom of the dialog box there is a group of options related to obtaining the averages of displacements. These options will affect the display and save file options as described above. This section of the dialog box provides the following options

- Displacement average in x and/or y directions. The displacement vector is projected along the standard x-y directions of the pixel grid and each component, or both, is obtained. If only one direction is chosen the output value for the other direction is set to zero.
- Then the average of each or both components is calculated along the directions defined by either the u or the v vectors. The average value is then output to every row of the output file.

Filters *Filters*

This menu consists of three menu-items, each being a different filter in reciprocal space.
They are:

Wiener Filter

Background Subtraction Filter

Threshold Filter

These routines are used to enhance the image of crystalline structures in the presence of amorphous noise. They calculate the Fourier transform of the original image and analyze the power spectrum through the histogram in order to make an estimate of the magnitude of the noise in the image. For each radius the histogram of the amplitudes of the Fourier coefficients is calculated and an estimate of the mean “noise” amplitude as a function of the reciprocal lattice vector is produced. The estimate of the noise can be used to construct a “Wiener Filter”, a “Background Subtraction Filter” or a “Threshold Filter”. For a further description of the filters, see Appendix 1.

All the filters are based on the idea that the combination of thin crystalline material together with an amorphous surface layer produces to a close

approximation two additive images. One of the crystalline material and one of the amorphous. Thus their Fourier transforms are additive as well.

The Wiener Filter uses the estimate of the noise to construct an optimal filter, multiplies the Fourier transform of the image with the filter and performs an inverse Fourier transform.

The “Background Subtraction Filter” reduces the amplitudes of each of the Fourier transform coefficients by subtracting out the average amplitude assigned to the noise. One can show mathematically that this filter is similar to the Wiener filter, but that it produces a somewhat stronger damping of the signal when the signal to noise ratio is low.

The threshold filter simply sets to zero all Fourier coefficients that have amplitudes less that the mean noise (plus some fraction of the standard deviation).

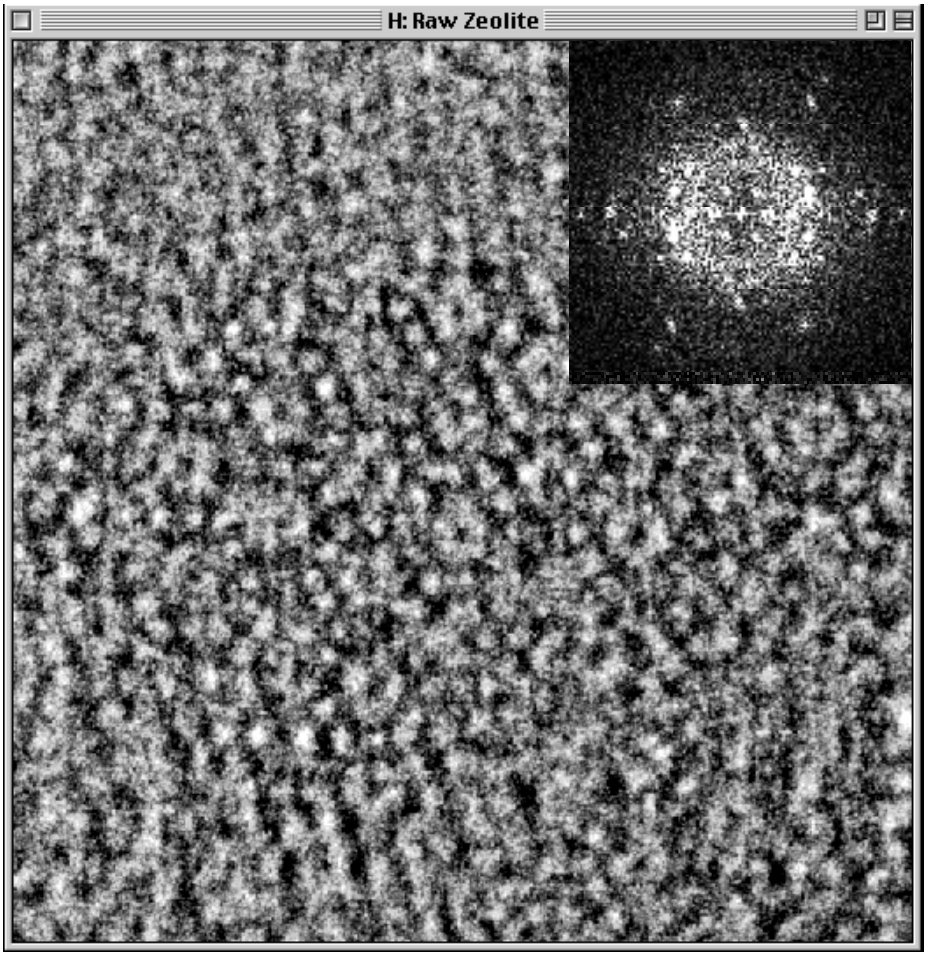
They take as input:

- The front image

They output:

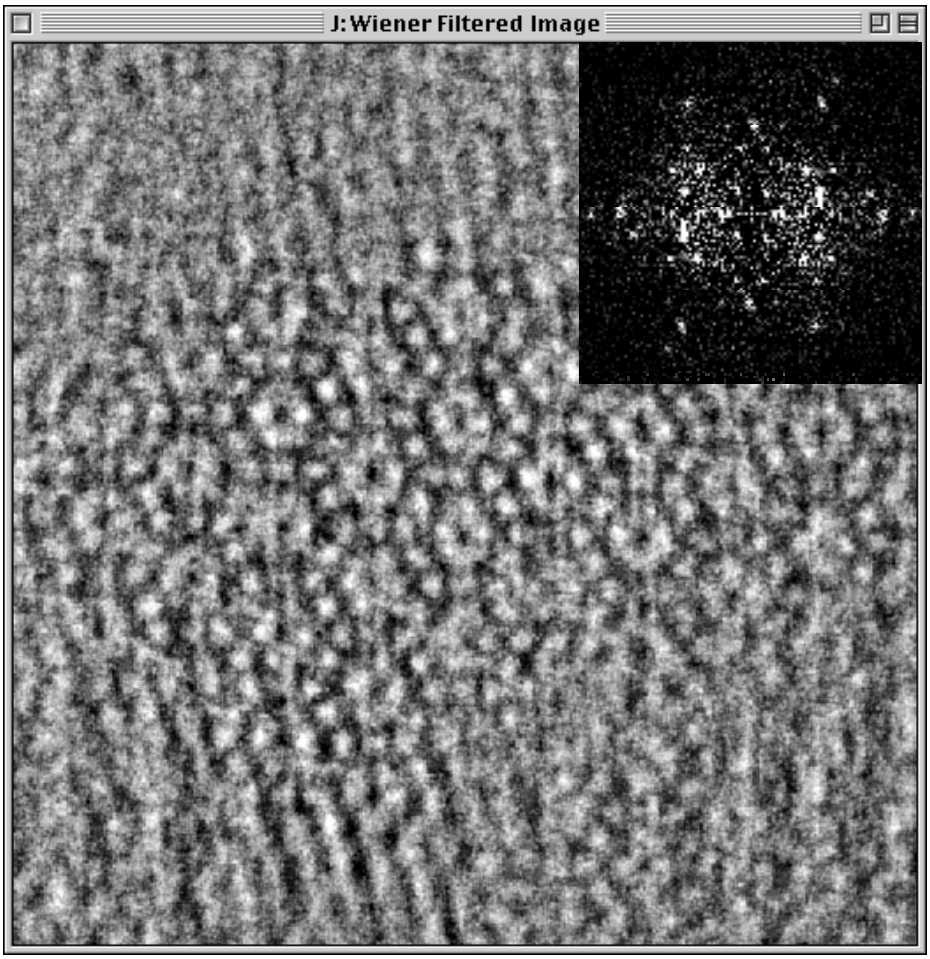
- An enhanced image *or*
- A line plot depicting the rotational average of the FFT of the image *and*
- A line plot depicting the standard deviation from the average of the FFT of the image

Example of the use of the filters on an image of Zeolite



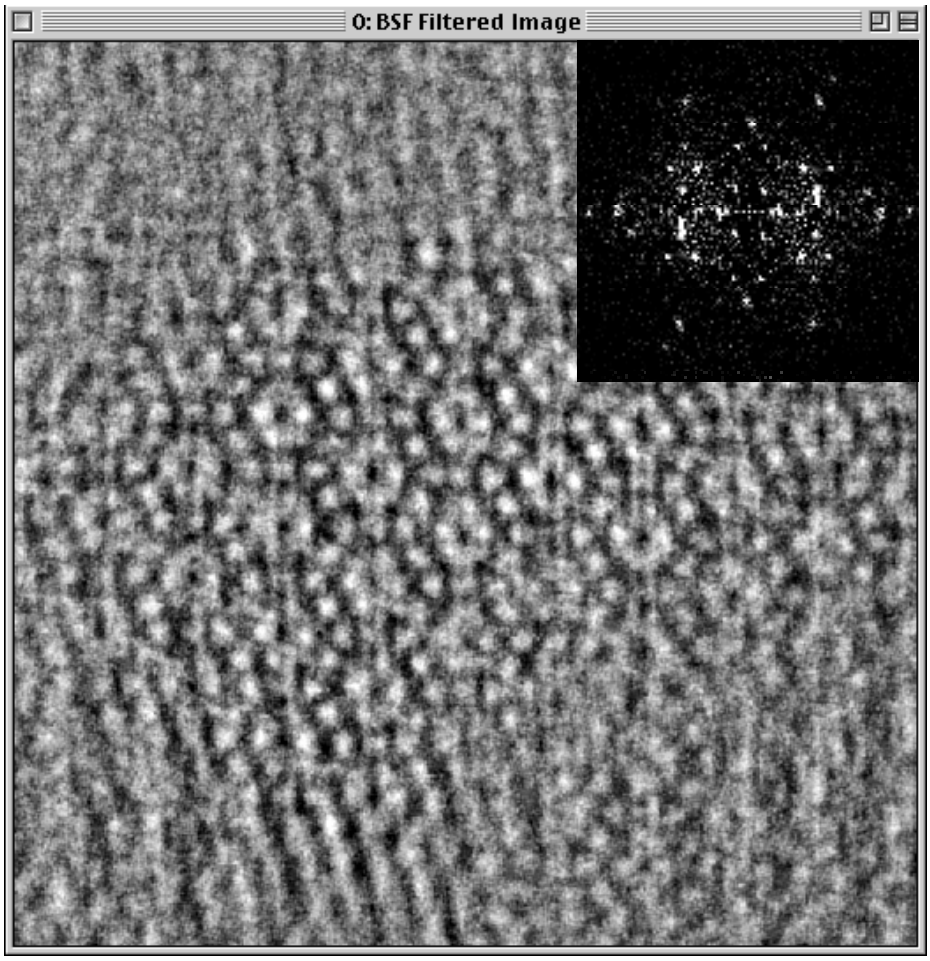
The above is the raw experimental image. The Fourier transform of the raw image is shown in the upper right corner of the image.

The result after using the Wiener Filter

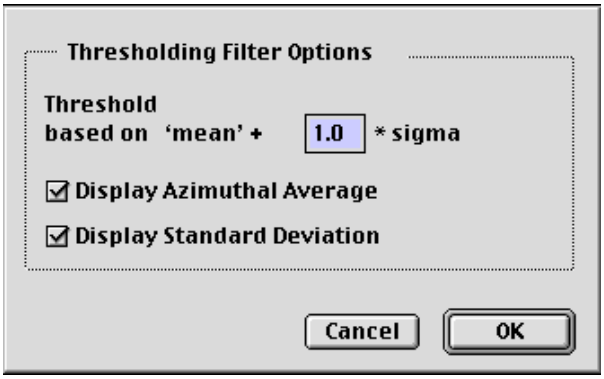


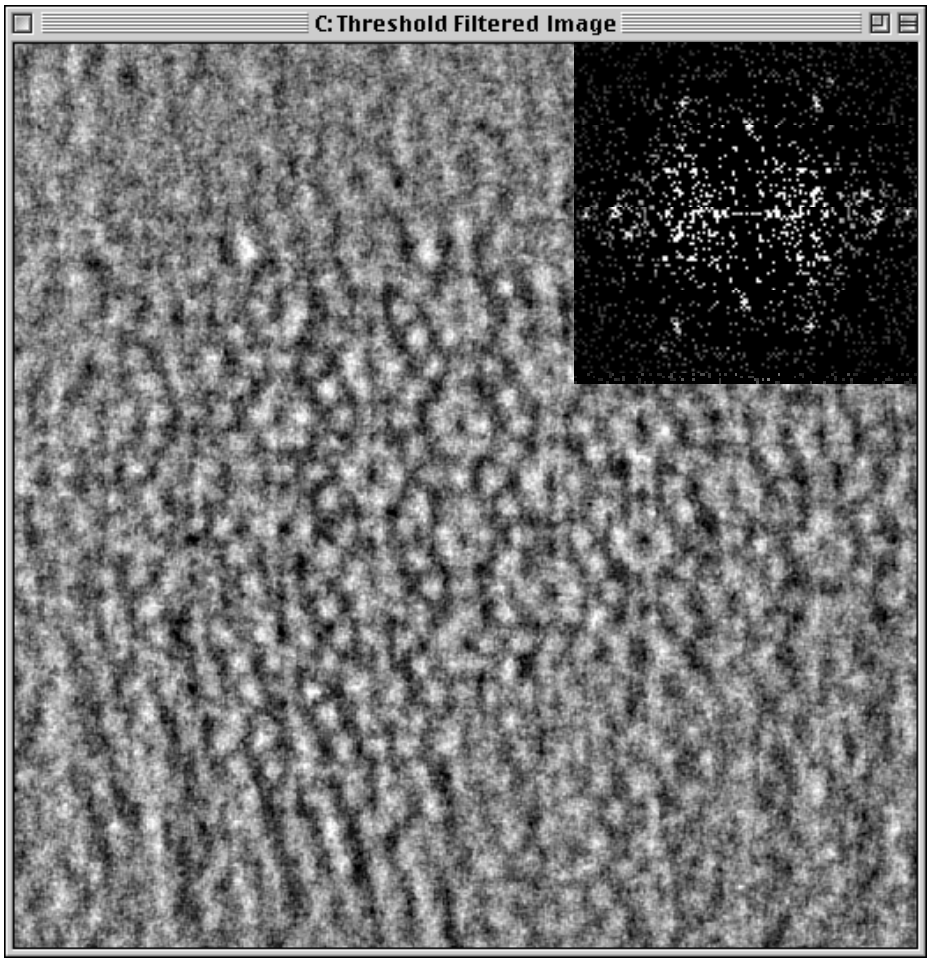
The success of the routines depends on the nature of the image, the amount of crystalline to amorphous material, the degree of overlapping of amorphous and crystalline material and the general thickness of the material.

The result of the Background Subtraction Filter



Applying the threshold filter




C: Threshold Filtered Image

Template Matching

Template Matching

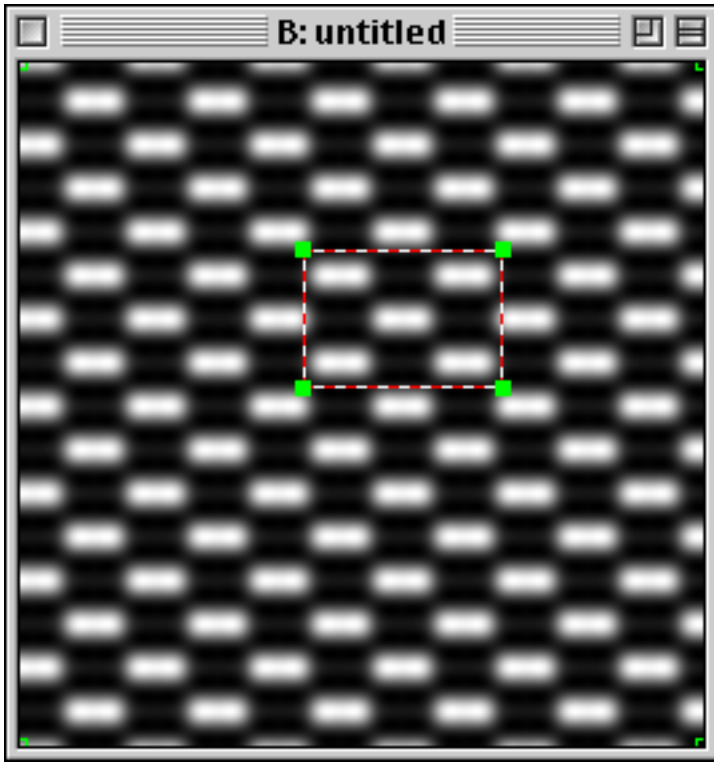
This routine calculates the Cross Correlation Coefficient (CCC) map between two images.

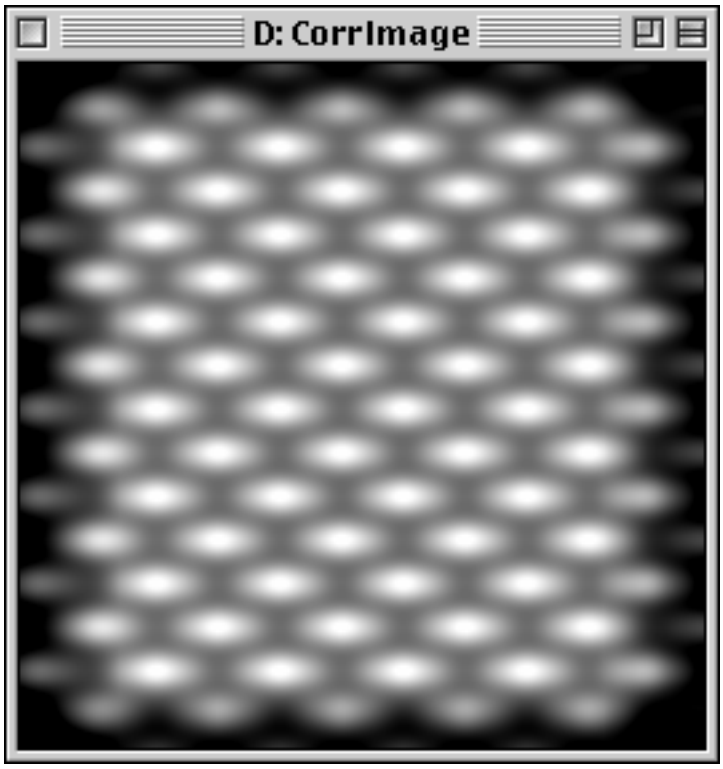
It takes as input:

- A work image
- A template image

It outputs:

- A CCC map which is an image of the same size as the work image, of type Real 4.





The template is scanned across the work image column by column and row by row. For each position of the template, the CCC with the underlying work image region is obtained.

The higher the coefficient the more similar the template is to the image at that position. The program then produces an output image whose intensities map the similarity between work image and template. The resulting image can then be thresholded or have its peaks found, to automatically locate patterns similar to the template.

A simple way of interpreting this operation is to regard the template and the underlying part of the work image as unidimensional vectors and obtain their dot product. The CCC corresponds to the cosine of the angle between the two vectors.

The routine can run in two modes:

Non-Normalized
In this mode the programs calculates the CCC directly from the original intensity values of the work image and template. The range of values for the CCC in this case is from 0 to 1 for typical images that have only posi-

tive values. This mode is fastest and should be used whenever a simple location of similar patterns is needed. However, no comparison based on CCC values obtained with different work images and/or templates is possible, unless all images have been normalized a priori.

Normalized
In this mode the programs calculates the CCC after subtracting the average from both work image and template. The average of the template is a fixed number. However, the average of the work image is calculated for the region underneath the template at each scan position. This value then has to be obtained for each position of the scan, increasing the computation time substantially. The range of values for the CCC in this case is from -1 to 1 even for positive pixel images. Even though this mode is slower, it provides values which do not depend on the average intensities of the work image and template. It is then suitable for comparing values obtained from different combinations of work image and template.

The only dialog presented by the program refers to choosing the two images that will be correlated. The order of the images is always irrelevant but there are some subtleties regarding the presence of selections in one or both images:

- To correlate an image with a part of itself, this part must be marked using the selection tool. In the “two images dialog” the user should choose the same image in both places. The program will automatically correlate the whole image with the selected region.

- When correlating two independent images, the program will check for selections in each image and will correlate only the selections. If only one image has a selection, the program will correlate the selection with the whole other image.

The program needs to create temporary scratch images. If there is not enough memory, the message “Unable to create scratch image” will appear. Try closing some images before proceeding.

Because the calculation can take a long time, depending on the size of image and template and on the normalization mode, a special time bar is provided that gives you an estimate of remaining calculation time. The initial estimates are generally incorrect because the template is still not fully overlapping the image. The first estimate will probably be inordinately high, then the value will fall and start increasing until the template totally overlaps the image. From there on the estimated time will decrease monotonically.

Circular Linescan

Circular Linescan

This routine is a simple routine that outputs as a 1D line plot the values along a circle. The number of bins (datapoints) can be selected by the user and also the integration width over which the values are averaged.

It takes as input:

- an Image

- a circular annotation on the image determining the path for extracting the values.

It outputs:

- a 1D image containing the (averaged) values along the circular path.

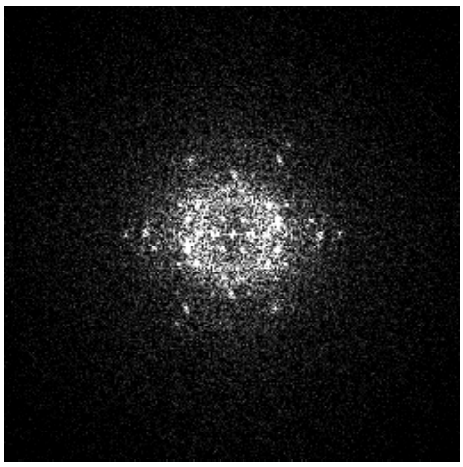
Azimuthal Average

Azimuthal Average

These routines calculates the azimuthal average of an image about its central pixel. The output depends on which menu-item is selected.

It takes as input:

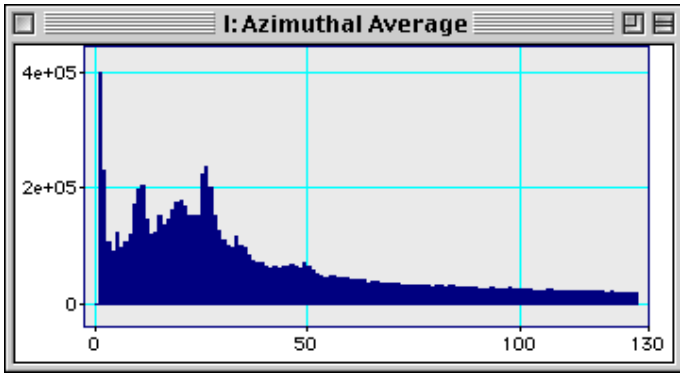
- The front image



It outputs:

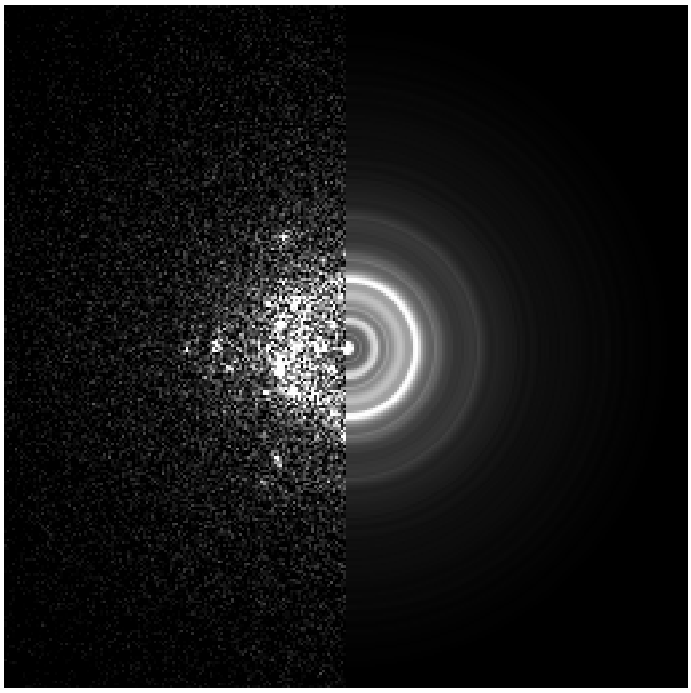
1D Image

- A one dimensional Line Plot depicting the rotational average.



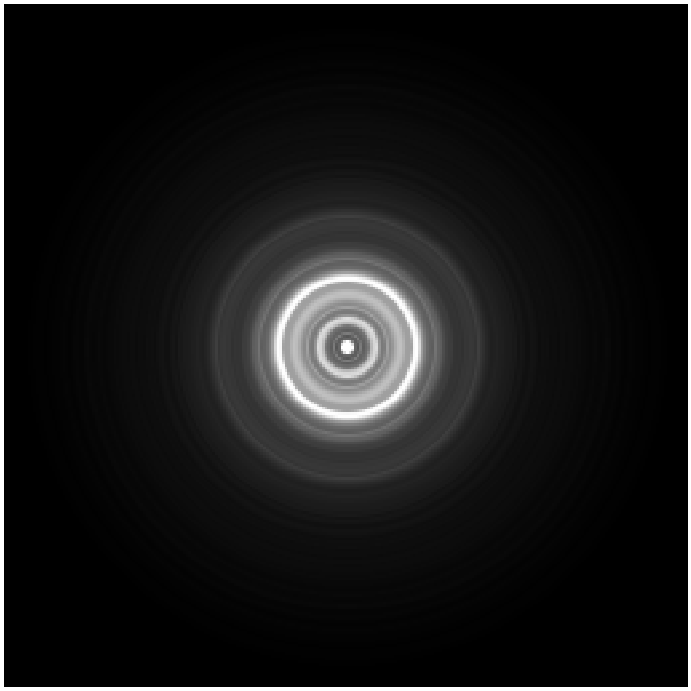
2D Image Split plane

- A two dimensional image showing the original data in the left half of the image and the azimuthal average in the right half.



2D Image

- - A two dimensional image which shows the azimuthal average as a ring pattern. This is the same as the 2D Image Split Plane with the average replacing the original data in the left half.



A typical use of this routine is in averaging diffractograms or diffraction patterns. Diffractograms, which are calculated through the FFT routine, have their center automatically positioned at the central pixel of the image. Diffraction patterns must have their central spot translated to the central pixel of the image before the rotational average is obtained.

Average / Extract Motif

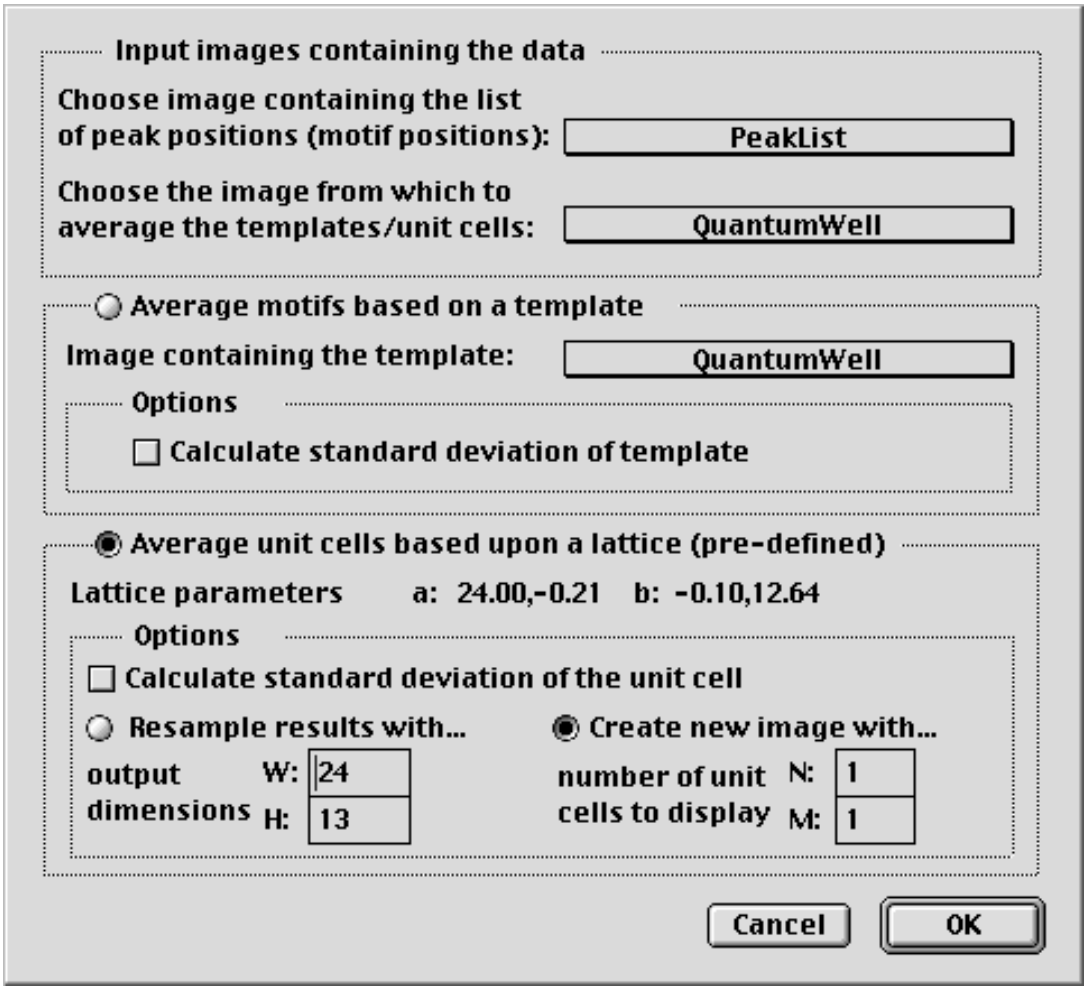
Average / Extract Motif

This routine is used to average many occurrences of a given template in an image. The actual actions and output depend on the various options selected by the user.

It takes as input three of the following:

- a Peaklist Image
- a Template Image
- a Lattice Image

- an Image from which to extract the information (referred to as Work Image)



It outputs on of the following:

- an average template image.
  - optionally a standard deviation image of the template.

- an average Unit cell (possibly resampled)
  - optionally a standard deviation image of the unit cell (possibly resampled)
- a new image containing a given number of averaged unit cells
  - optionally a standard deviation image of given number of unit cells

a) Average motifs based on a template
Based upon a PeakList (typically that obtained from peak finding a cross-correlation image-map obtained from running Template Matching), the routine will extract selections of the size of the template from the work image. The selections which are centered around the peak positions are added together to give an average template.

b) Average unit cells based upon a lattice
Based upon a PeakList defining the positions of similar motifs and a lattice defining the size and orientation of the unit cells within the work image, the data contained within the unit cell is averaged over recurring motifs. The standard deviation for each pixel is also optionally computed. Two different results are produced depending on the following options.

i) Resample results with...

The initial values are those obtained from the size of the lattice parameters to the nearest integer value. The output is always a rectangular image containing one unit cell data and optionally the standard deviation. If the motif (lattice) does not have a 90° angle, the image will be distorted. However, if the image is later to be compared to a simulated image within an image simulation program like MacTempas, this rectangular image corresponds to what is computed within MacTempas, before the image is being displayed and the angle is taken into account.

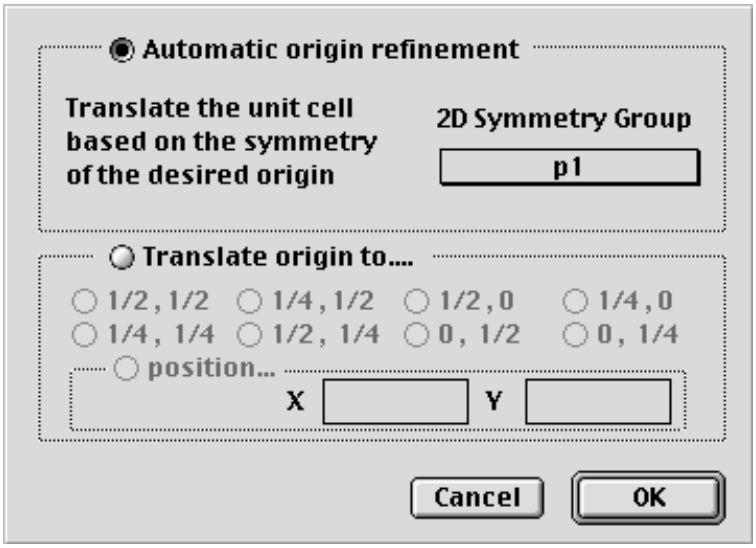
ii) Create New Image with...

A new image is created which consists of the requested number of unit cells. The image will show the average unit cells with the same orientation as the original image.

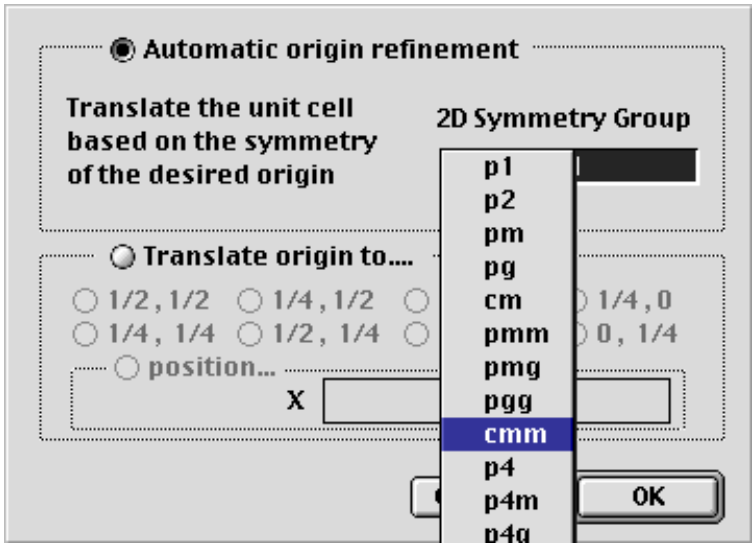
Change Motif Origin

Change Motif Origin

This routine is used to change the origin of the average unit cell.



a) Automatic origin refinement
If the unit cell is sampled onto a grid with dimensions that are powers of two, the routine can be used to find the origin which satisfies most closely the requested symmetry.



b) Translate origin to...
This will move the origin to one of the selected positions within the image.

The procedure assumes that the input image is a periodic function in X and Y and thus when it creates the output image will periodically extend the input image.

Input:

- an Image

Output:

- a new image which reflects the shifted origin.

Holography Routines

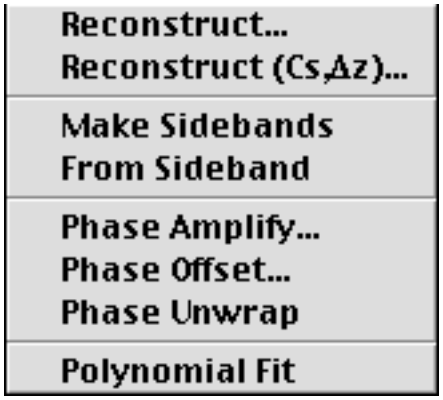
Holography Routines

An electron hologram is a fringe modulated image containing the amplitude and phase information of an electron transparent object. These routines "reconstruct" the amplitude/phase images allowing user control of various parameters. In addition, high resolution holograms can be reconstructed by means of the contrast transfer function of the objective lens to backcalculate the object exit wavefunction. Holograms are considered to have the following form:

$$I(r) = 1 + A(r)*A(r) + 2*A(r)*\cos(P(r) + 2\pi q(r))$$

where I is the image intensity in real-space (r), A the amplitude, P the phase and q the hologram carrier fringe modulation.

These routines operate on hologram images of size 2^n x 2^n where n is an integer. Also a "good" hologram has at least 4 pixels/fringe. The displayed phase images are "wrapped" between $-\pi$ to $+\pi$ radians. Due to the large size of the images used in reconstructing a hologram, the memory requirements for a 1024x1024 hologram image is about 60 MB. The routines require at least a Macintosh Quadra, preferably a PowerMac.



Reconstructions...

Reconstructions...

Reconstruct

Reconstruct

takes the hologram (front image) and applies a Hanning window. The Fourier transform is then calculated and the exact position of the sideband determined. The user may use either the sideband or its phase conjugate to reconstruct the hologram. The user controls the size of the masking aperture as well as the aperture shape (circle or square). The amplitude and (wrapped) phase images are then calculated and displayed. Two line plots of the phase are displayed, the x-profile and the y-profile. The user may iteratively tilt the comparison wave used to calculate the phase as well as change the phase offset.

Notes: the user-specified aperture size S is determined as a circle of radius S or a square with edge 2S where S is by default half the distance (in pixels) from the auto-correlation to the sideband. The dialog also contains the proper value for a one-third sized aperture. The aperture is centered on the chosen sideband. The tilts of the comparison wave are real numbers, typically values from -5 to 5 are adequate, however any number can be used. The routine stores the tilt values for user reference, resetting the x, y and offset values to zero recovers the original phase image.

Reconstruct (Cs,Δz)

Reconstruct (Cs,Δz)

takes a high resolution hologram and exactly determines the position of the sideband as before. The user may then specify the contrast transfer function (CTF) parameters of the microscope used to record the hologram. The routine removes the effect of the CTF (in the form of exp(-iX[u])*envelope terms) to calculate the object exit wavefunction. The amplitude and (wrapped) phase images are calculated and displayed along with the phase x- and y-profiles. The user may iteratively change the parameters of the CTF to recalculate the amplitude and phase images. Note: the hologram must be calibrated in either Å or nm units, else the routine exits. The amplitude and phase images are calibrated in nanometer units.

Sidebands...

Sidebands...

Make sidebands will calculate the Fourier transform of the hologram and Hanning window. A complex image is displayed. The Hanning window is stored as a hidden image. The user is then free to manipulate the pixel data, for example, masking the sidebands with apertures of arbitrary shape and position.

From sideband assumes a complex image containing a single sideband (i.e., no auto-correlation or conjugate sideband). The exact sideband position is determined and the amplitude and phase images calculated and displayed as well as the x- and y-profiles of the phase. The user may iteratively tilt the comparison wave used to calculate the phase and change the phase offset.

Phase data...

Phase data...

Phase amplify takes any phase image. The routine calculates the N-times amplified phase according to user-specified factor N. The user may also offset the phase from -π to +π. Note: N=1 amplification is meaningless.

Phase offset

Phase offset

takes a wrapped phase image and recalculates according to the user-specified offset to change the “baseline” of the phase. The phase data remains wrapped from -π to +π.

Phase unwrap

Phase unwrap

takes a wrapped phase image and removes the discontinuous jumps that result from the original phase wrapping (from -π to +π). The phase is unwrapped from a user-specified origin. Note: rapid jumps near image edges are problematic... works best on the center 7/8 of the image.

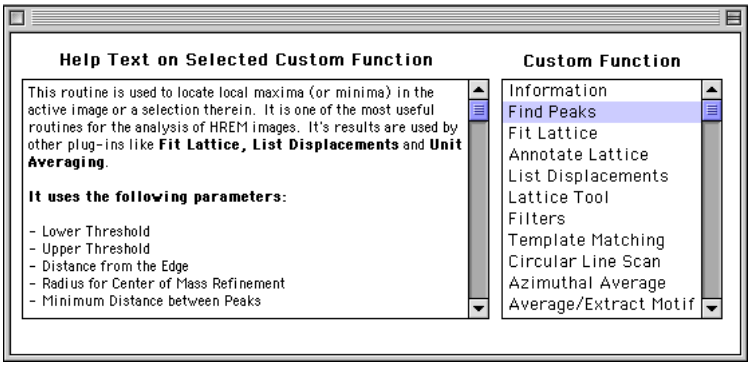
Polynomial fit

Polynomial fit

takes any image and attempts to fit the data to a user specified Nth order polynomial (max N of 6). The user may specify the number of iterations (typically 100) to allow for fitting convergence. Note: discontinuous (wrapped) phases can not be fitted.

NCEM Help

NCEM Help



Help produces a window which allows you to access the information in this manual from within the program.

Appendix 1. Theory of the noise reduction filters

Appendix 1. Theory of the noise reduction filters

The following text is a modified versions of text appearing in the article

" Optimal and Near-Optimal Filters in High Resolution Electron Microscopy ", R. Kilaas, Journal of Microscopy, Vol. 190, Pts 1/2, April/May 1998, pp. 45-51.

1. Introduction

Most images recorded in High Resolution Transmission Electron Microscopy (HRTEM) show the presence of amorphous layers, such as native oxide, contamination or support film. The amorphous layer degrades the information which comes from the crystalline material of interest and lowers the signal to noise ratio. What is referred to as signal and noise in this case are actually two signals, one coming from the crystalline region of the specimen and the other coming from the amorphous layer(s).

The signal which is recorded on film, image plate or slow scan CCD camera is a function of the interaction of the electrons with the specimen and cannot in principle be separated into components from an amorphous region and a crystalline region. However, as a first approximation, the effect of the amorphous material is to add the image of the amorphous material to the image of the crystalline material. Mathematically, this can only be shown within the validity of the Weak Phase Object Approximation (WPOA). For thin specimens and using the WPOA (Cowley & Moodie, 1957), the electron wavefunction at the exit surface of the specimen can be written

$$\Psi(\mathbf{r}) = 1 + i\sigma(V_c(\mathbf{r})t_c + V_a(\mathbf{r})t_a) \qquad \text{real space} \qquad (1)$$

or

$$\Phi(\mathbf{u}) = \delta(\mathbf{u}) + i\sigma(V_c(\mathbf{u})t_c + V_a(\mathbf{u})t_a) \qquad \text{reciprocal space} \qquad (2)$$

where s is the interaction constant, t_c and t_a are the thicknesses of the crystalline and the amorphous material, and V_c and V_a are the two respective projected potentials.

The effect of the objective lens on the electron wavefunction is to add in reciprocal space a phase factor $X(\mathbf{u})$ such that to the first approximation the Fourier transform of the recorded intensity is

$$F(\mathbf{u}) \approx 2\sigma(V_c(\mathbf{u})t_c + V_a(\mathbf{u})t_a)\sin X(\mathbf{u}) \equiv F_c(\mathbf{u}) + F_a(\mathbf{u}) \quad (3)$$

with the corresponding real space representation

$$I(\mathbf{r}) \approx I_c(\mathbf{r}) + I_a(\mathbf{r}) \quad (4)$$

The components $F(\mathbf{u})$ are complex and because the image is real, $F^*(\mathbf{u}) = F(-\mathbf{u})$.

The expression above does not include the effects of partial coherence in which the recorded image is a sum of incoherent images from electrons with a spread of energies and incoming directions. However, including partial coherence does not change the decomposition of the recorded signal into the two components, one crystalline and one amorphous, provided that non-linear terms are neglected in the image formation (O'Keefe 1979). The weak phase object approximation is usually valid for the amorphous layer and also applies in many cases for the crystal for the thicknesses where the amorphous layer shows a large effect on the image quality.

The two signals have distinctly different characteristics in the frequency domain. The amorphous contribution is spread out in reciprocal space with amplitudes that depend on the magnitude of the reciprocal vector and can be considered independent of the azimuthal angle except for random variations. The signal from the crystalline material is peaked around specific spatial frequencies which correspond to spacings between atomic planes. Thus while the information in real space is inseparable, the two signals can to a large extent be separated in the frequency domain.

The aim of this paper is to outline automatic procedures for estimating the signal $F_c(\mathbf{u})$ from $F(\mathbf{u})$ (obtained from the Fourier transform of the recorded signal $I(\mathbf{r})$) and thus obtaining an estimate of the image $I_c(\mathbf{r})$ which would have been recorded in the absence of the amorphous material.

2. Mathematical derivation of the filters

2.1. Optimal Filter (Wiener)

The estimate for the signal $F_c(\mathbf{u})$ is written as:

$$F_c^{est}(\mathbf{u}) = F(\mathbf{u}) \cdot \Theta(\mathbf{u}) \tag{5}$$

where we seek to optimize the solution such that the summed square differences between the true signal $F_c(\mathbf{u})$ and the estimate $F^{est}{}_c(\mathbf{u})$ is minimized.

This is expressed as minimizing the quantity

$$\sum_{\mathbf{u}} \left| F_c^{est}(\mathbf{u}) - F_c(\mathbf{u}) \right|^2 = \sum_{\mathbf{u}} \left| (F_c(\mathbf{u}) + F_a(\mathbf{u})) \cdot \Theta(\mathbf{u}) - F_c(\mathbf{u}) \right|^2 \tag{6}$$

The expression above is minimized by minimizing each term. Setting the derivative with respect to $\Theta(\mathbf{u})$ equal to zero gives

$$\Theta(\mathbf{u}) = \frac{\left| F_c(\mathbf{u}) \right|^2}{\left| F_c(\mathbf{u}) \right|^2 + \left| F_a(\mathbf{u}) \right|^2} \tag{7}$$

The filter above is well known and is referred to as the Wiener filter (Rabiner & Gold, 1975). In the derivation of the Wiener filter, it is important to note that the quantities $F_c(\mathbf{u})$ and $F_a(\mathbf{u})$ are treated as uncorrelated. Thus there is an underlying assumption that there is no correlation between the signals from the crystalline and the amorphous materials. The degree of correlation between the estimates of the two signals will vary depending on the thickness of the amorphous and crystalline part (Hÿtch & Chevalier, 1994), but will be considered to a first approximation to be negligible. The Wiener filter and variants thereof, including the Cannon filter (Cannon, 1977), and their use in HREM have been discussed in detail by Marks (Marks, 1996).

In order to determine $\Theta(\mathbf{u})$ we need estimates for both quantities $|F_c(\mathbf{u})|^2$ and $|F_a(\mathbf{u})|^2$.

"Because the optimal filter results from a minimization problem, the quality of the results obtained by applying the optimal filter differs only from the true optimum by an amount that is second order in the precision to which the filter is determined" (Press et al., 1986). Thus excellent results are often obtained by fairly crude estimates of the filter.

Writing

$$|\mathrm{F}_c(\mathbf{u})|^2 + |\mathrm{F}_a(\mathbf{u})|^2 \approx |\mathrm{F}(\mathbf{u})|^2 \tag{8}$$

gives as the estimate for the optimal filter

$$\Theta_{WF}(\mathbf{u}) \approx \frac{|\mathrm{F}(\mathbf{u})|^2 - |\mathrm{F}_a(\mathbf{u})|^2}{|\mathrm{F}(\mathbf{u})|^2} \tag{9}$$

so that, together with the Fourier transform of the recorded signal, the only quantity that needs to be determined is $|\mathrm{F}_a(\mathbf{u})|^2$. Section 3 will discuss how to determine $|\mathrm{F}_a(\mathbf{u})|^2$ from $|\mathrm{F}(\mathbf{u})|^2$.

2.2. Average Background Subtraction Filter (ABSF)

By writing the Wiener filter as

$$\Theta_{WF}(\mathbf{u}) \approx \frac{|\mathrm{F}(\mathbf{u})|^2 - |\mathrm{F}_a(\mathbf{u})|^2}{|\mathrm{F}(\mathbf{u})|^2} = \frac{(|\mathrm{F}(\mathbf{u})| - |\mathrm{F}_a(\mathbf{u})|)(|\mathrm{F}(\mathbf{u})| + |\mathrm{F}_a(\mathbf{u})|)}{|\mathrm{F}(\mathbf{u})|^2} \tag{10}$$

one can define a filter $\Theta_{\mathrm{ABSF}}(\mathbf{u})$ as follows:

$$\Theta_{WF}(\mathbf{u}) \approx \frac{|\mathrm{F}(\mathbf{u})| - |\mathrm{F}_a(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})|} \cdot \frac{|\mathrm{F}(\mathbf{u})| + |\mathrm{F}_a(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})|} \equiv \Theta_{ABSF} \cdot \frac{|\mathrm{F}(\mathbf{u})| + |\mathrm{F}_a(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})|} \tag{11}$$

This gives

$$\Theta_{ABSF}(\mathbf{u}) = \frac{|\mathrm{F}(\mathbf{u})| - |\mathrm{F}_a(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})|} \tag{12}$$

with the relationship between the two filters as:

$$\Theta_{ABSF} = \Theta_{WF} \cdot \frac{|\mathrm{F}(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})| + |\mathrm{F}_a(\mathbf{u})|} \tag{13}$$

The asymptotic behavior of the two filters are the same, with the filters being zero when $\mathrm{F}(\mathbf{u}) = \mathrm{F}_a(\mathbf{u})$ and close to 1 when $\mathrm{F}_c(\mathbf{u}) >> \mathrm{F}_a(\mathbf{u})$. The background subtraction filter results in a little stronger filtering in the intermediate regime, but as will be seen from the examples shown in Section 4, the two filters give comparable results.

In the above the filter $\Theta_{\mathrm{ABSF}}(\mathbf{u})$ has been derived from the filter $\Theta_{\mathrm{WF}}(\mathbf{u})$. In fact the historical background of the filter $\Theta_{\mathrm{ABSF}}(\mathbf{u})$ is that it was introduced and used at the NCEM for a number of years before its relationship to the Wiener filter was understood. The background subtraction filter is incorporated into the NCEM image processing extension (Kilaas & Paciornik, 1995) for the software package Digital Micrograph (Gatan, Inc.). The ABS filter should not be confused with another filter of similar name (Sattler & O'Keefe, 1987) which is a subjective filter, requiring the user to identify reflections associated with the crystalline material and to construct a background by replacing strong peaks with Fourier components whose phases are random and whose amplitudes are taken from regions near the peaks.

The average background subtraction filter has a simple geometric description from which it derives its name. If one writes the result of the signal as

$$\mathrm{F}_c^{est}(\mathbf{u}) = \mathrm{F}(\mathbf{u}) \cdot \Theta_{ABSF}(\mathbf{u}) = \mathrm{F}(\mathbf{u}) \cdot \frac{|\mathrm{F}(\mathbf{u})| - |\mathrm{F}_a(\mathbf{u})|}{|\mathrm{F}(\mathbf{u})|} = \hat{\mathrm{F}}(\mathbf{u}) \cdot (|\mathrm{F}(\mathbf{u})| - |\mathrm{F}_a(\mathbf{u})|) \tag{14}$$

then the vectorial representation of the resulting Fourier component is obtained from the Fourier component of the recorded signal minus a vector in the same direction whose length is given by the estimated amplitude of the signal $\mathrm{F}_a(\mathbf{u})$.

3. Estimating the background

The signal from the amorphous background is spread out in reciprocal space with an amplitude that depends on the radial frequency |u|, and is independent of direction, except for random variations. Thus in order to estimate the amplitude $|F_a(\mathbf{u})|$ it is necessary to find the average Fourier amplitude for a given radial frequency $|\mathbf{u}|$, excluding in principle any contributions from the signal $F_c(|\mathbf{u}|)$. However, the signal $F_c(\mathbf{u})$ is localized in reciprocal space and for any given radius $|\mathbf{u}|$ more than a few pixels away from the center, $F_c(\mathbf{u})$ only corresponds to a limited number of sampling points along the curve $|\mathbf{u}|$ = constant. In addition, the signal $F_c(\mathbf{u})$ usually is much larger than $F_a(\mathbf{u})$ and can often be distinguished from $F_a(\mathbf{u})$ based upon its magnitude. It turns out that the results obtained by applying the filters are not very sensitive with respect to how the average of the background signal is computed. Estimating the background signal by averaging over the entire circle $|\mathbf{u}|$=constant, often give good results even though the estimated magnitude of the background will be higher than the actual value when crystalline peaks are present along the circle. However, the background can be better estimated from the histogram of the amplitudes. The histogram will separate the background signal from any large crystalline contribution, with the majority of the pixel values coming from the background signal due to the amorphous material. Thus the average amplitude $|F_a(|\mathbf{u}|=\text{constant})|$ can be estimated by determining the histogram of amplitudes along the circle $|\mathbf{u}|$ = constant and computing the mean of the histogram, excluding outliers which will reduce any contribution from the signal $F_a(\mathbf{u})$. A typical histogram of Fourier amplitudes for $|\mathbf{u}|$=constant when crystalline peaks are present consists only of a relatively small number of pixels that correspond to the crystalline material and the histogram distribution becomes bi-modal with the majority of the pixels contributing to the lower cluster of peaks.

The average background signal is computed for every radius in order to derive the estimate for $|F_a(\mathbf{u})|^2$. Subsequently $|F_c(\mathbf{u})|^2$ is estimated from

$$\left|F_c(\mathbf{u})\right|^2 \approx |F(\mathbf{u})|^2 - \left|F_a(\mathbf{u})\right|^2 \approx |F(\mathbf{u})|^2 - \left\langle \left|F_a(|\,\mathbf{u}\,|)\right|^2 \right\rangle \qquad (15)$$

in order to compute the filters $\Theta_{WF}(\mathbf{u})$ and $\Theta_{ABSF}(\mathbf{u})$

4. Discussion

With any type of filtering, there is a danger that the filtered image no longer represents the raw data in any meaningful way. However, there are many times when one makes intelligent guesses with respect to the nature of the recorded signal and the imaging conditions and proceeds accordingly. The underlying assumptions in this work is that the Fourier transform of the recorded image can be treated as a sum of two uncorrelated signals. Neither statement will strictly be true, but as a starting point, they are presumed to hold to a first approximation. The results obtained by applying the two filters need to be compared to the actual data to ensure that they are free of obvious artifacts. Obvious artifacts such as extension of crystalline features into non-crystalline regions indicate that the filter is inappropriate and often this is caused by too much filtering resulting from an over-estimation of the noise. Since the signal ("crystalline image") is unknown for an experimental image, it is difficult to assess quantitatively the effects of the filters on experimental data. However, for simulated data with known crystalline material and known noise, the results can be compared quantitatively. Tests show that depending on the size of the image and how the estimate of the noise relates to the true noise, the improvements in signal to noise lie in the range 3-7 (Marks, 1996).

Both the Wiener filter and the average background subtraction filter are created automatically from the data by estimating the power spectrum of the signal due to the amorphous background, assuming that, apart from random variations, the signal only varies with radial frequency $|\mathbf{u}|$. The Wiener filter gives the optimal estimator for the signal $F_c(\mathbf{u})$ in a least squares sense. The background subtraction filter results in a slightly heavier damping of frequencies where the "signal" ($F_c(\mathbf{u})$) and "noise"($F_a(\mathbf{u})$) are comparable in magnitude, but otherwise gives results very similar to the Wiener filter. Given the availability of both filters, it is clear that the Wiener filter should be chosen over the background subtraction filter. However, because the background subtraction filter has been in use by many scientists for a number of years, it is important to provide information on its applicability and its relationship to the Wiener filter. Both filters operate automatically. No choice is offered to the user, and no further assumptions are made, apart from those built into the algorithm. If properly applied, they can assist in extracting useful information from HRTEM images when the image consists of crystalline material in conjunction with an amorphous layer.

References

Cannon, T.M. (1977) Digital Image Deblurring by Nonlinear Homomorphic Filtering, Ph.D Thesis, Computer Science Department, University of Utah, Salt Lake City, UT

Cowley, J.M. & Moodie, A.F. (1957) The Scattering of Electrons by Atoms and Crystals. I. A New Theoretical Approach, Acta Cryst., 10, 609-619

Gatan, Inc, 6678 Owens Drive, Pleasanton, CA 94588, USA.

Hÿtch, M.J, and Chevalier, J.P. (1994) The effect of Amorphous Surface Layers on HREM Images of Crystalline Material, Proceedings of ICEM, Paris, 367-368

Kilaas, R. & Paciornik, S. (1995) The NCEM public domain software library of extensions to Digital Micrograph. Proc. of MSA 46, 628-629

Marks, L.D. (1996) Wiener-filter enhancement of noisy HREM images, Ultramicroscopy 62, 43-52

O'Keefe, M.A. (1979) Resolution-damping functions in non-linear images, Proc. of EMSA 37, 556-557

Paciornik, S., Kilaas, R., Turner, J., Dahmen, U. (1996) A Pattern Recognition Technique for the Analysis of Grain Boundary Structure by HREM. Ultramicroscopy, 62, 15-27

Press, W.H. et al (1986) Numerical Recipies, The art of scientific computing. Cambridge University Press 1986 ISBN 0 521 30811 9, p. 419

Rabiner, L.R. & Gold, B. (1975), Theory and application of digital signal processing Englewood Cliffs, N.J.: Prentice-Hall

Sattler, M.L. and O'Keefe, M.A. (1987) Atomic Structure Analysis of Small Particles Supported on Amorphous Material, Proc. of EMSA 45, 104-105